

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

Zadání bakalářské práce

Patrik Herák

Student:

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: LOGIS a.s.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti
 - c) Zvolený postup řešení zadaných úkolů
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vedl odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Mgr. Pavla Dráždilová, Ph.D.**

Konzultant bakalářské práce: Ing. Pavel Kotla

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prehlasujem, že som túto bakalárskú prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

V Ostravě 2. Máj 2014



.....

Rád by som na tomto mieste poďakoval svojmu konzultantovi Ing. Pavlovi Kotalovi za vedenie bakalárskej praxe. Ďalej by som chcel poďakovať Ing. Zbyňkovi Ondryášovi, ktorý mi taktiež veľmi pomáhal vo firme. Za odborné vedenie a pripomienky by som chcel poďakovať vedúcej mojej bakalárskej praxe Mgr. Pavle Dráždilovej, Ph.D. V neposlednom rade by som chcel poďakovať za podporu mojej priateľke a celej jej rodine, pretože bez nich by táto práca nevznikla.

Abstrakt

Táto bakalárska práca bola vykonaná formou odbornej praxe vo firme LOGIS a.s. Firma sa zameriava na tvorbu aplikácií pre plánovanie výroby v hutníckom a strojárenskom priemysle. Na praxi bola vykonávaná pozícia testera vyvíjaných aplikácií, ktorá bola spojená taktiež s tvorením testovacieho scenáru, dokumentácie, nahlasovania problémových častí aplikácií, navrhovaní riešení, atď. Neskôr bola využitá znalosť programovacieho jazyka Java a dopytovacieho jazyka SQL pri tvorbe aplikácii, ktorá vyhodnocuje výsledky DVP behov.

Kľúčové slová: logistika, java, prax, testovací scenár, dokumentácia, plánovanie, splniteľnosť objednávky

Abstract

This bachelor thesis was done in the form of individual professional practice in the company LOGIS a.s. The company focuses on creating applications aimed at planning of production in the steel and engineering industry. The position of software tester was carried out - this position was associated also with creating testing scenario, documentation, bug reporting, suggesting solutions, etc. The knowledge of programming languages Java and SQL was used in a process of developing an application, which evaluates results of DVP runs.

Keywords: logistics, java, practice, testing scenario, documentation, planning, order promising

Zoznam použitých skratiek a symbolov

GUI	– Graphical User Interface
KPI	– Key Performance Indicators
Trac	– Systém na zaznamenávanie chýb / zlepšení
DVP	– Detail Verification Process
LOP	– LOGIS Order Promiser
LMP	– LOGIS Master Planner
CSV	– Comma-separated values

Obsah

1	Úvod	4
2	Firma LOGIS a.s. a používané technológie	5
2.1	Firma LOGIS a.s. a jej aktivity	5
2.2	Charakteristika programu LOGIS Order Promiser	6
2.3	Detail Verification Process (DVP)	10
3	Priebeh praxi	11
3.1	Zoznámenie sa s firmou LOGIS a.s.	11
3.2	Harmonogram praxi	11
3.3	Zaučovanie do programu	13
3.4	Testovanie LOP	13
3.5	Web meeting	16
3.6	Tvorba testovacieho scenára	17
3.7	Testovanie užívateľských práv v LMA	18
3.8	Tvorba dokumentácie k softwaru LMP	18
3.9	Import štruktúry CUBE z CSV súboru	19
3.10	Tvorba softwaru pre vyhodnocovanie výsledkov DVP behu	20
4	Záver	23
5	Literatúra	24

Zoznam obrázkov

1	LOP Workflow	6
2	Pôvodné rozloženie GUI prvkov v LOP	8
3	Náhľad do CUBE v reálnom použití	8
4	Znázornenie CUBE ako stromu	9
5	Postupnosť, akou prechádzajú objednávky počas DVP procesu	10
6	Chybová hláška - No such category value.	14
7	Zmiznutie hlavného JFrame	15
8	Zlá štruktúra, ktorú LOP vyhodnotil ako správnu	20
9	Výsledný prehľad DVP výsledkov	22

Zoznam výpisu zdrojového kódu

1	Nastavovanie prostredí v LMA	18
2	Výraz pre obmedzenie práv na určité prostredie	18
3	Výsledný SQL dopyt pre spočítanie objednávok, ktoré vstupujú do DVP procesu	21
4	SQL dopyt pre získanie najnovšieho DVP behu	21
5	Výsledný SQL dopyt pre získanie počtu objednávok, ktoré sa vrátili z DVP procesu	22
6	Výsledný SQL dopyt pre získanie počtu objednávok, ktoré nemajú žiadne alokácie	22

1 Úvod

Bakalárska prax bola vykonávaná vo firme LOGIS a.s., ktorá sídli vo Frenštátě pod Radhoštěm. Táto firma sa zaoberá vývojom aplikácií pre plánovanie výroby v hutníckom a strojárenskom priemysle. Svoje produkty poskytuje hlavne zahraničným klientom (Amerika, Japonsko, Rusko ...). Vďaka tomu je všetka textová komunikácia vo firme v anglickom jazyku. Na praxi mi bolo umožnené pracovať ako tester programu LOGIS Order Promiser (tzv. LOP), ktorý pomáha odberateľskej firme zistiť (na základe vstupných požiadavok), či sú objednávky v ich systéme splniteľné v požadovanom dátume (kap. 2.2). Po dôkladnom zoznámení s programom mi bola zverená úloha na pomoc pri vývoji aplikácie, ktorá vyhodnocuje výsledky z DVP behov (kap. 2.3).

Súčasťou tejto práce bude nielen zoznámenie s firmou LOGIS a.s., ale aj zoznámenie s programom LOP a jeho časťami, s DVP procesom a náplňou mojej práce.

Od tejto praxe očakávam, že mi priblíži, aké to je pracovať pre takú firmu, využijem tu poznatky z môjho bakalárskeho štúdia na VŠB-TUO a taktiež získam nové skúsenosti a schopnosti v oblasti programovania a tímovej práce. V neposlednej rade vidím ako veľkú výhodu to, že všetka komunikácia vo firme prebieha v anglickom jazyku, takže si môžem zdokonaľiť tento cudzí jazyk.

2 Firma LOGIS a.s. a používané technológie

V tejto kapitole sa nachádza zoznámenie s firmou LOGIS a.s., hlavne s ich špecializáciou a prínosom pre odberateľa. Ďalej je tu opísaná základná charakteristika programu LOGIS Order Promiser, časť aplikácie nazývaná CUBE a dôležité serverové akcie využívané v programe. Nakoniec je okrajovo opísaný DVP proces, ktorý sa vo firme vyvíja.

2.1 Firma LOGIS a.s. a jej aktivity

LOGIS je dodávateľom expertných služieb a informačných technológií zameraných na zlepšovanie kvality riadenia a konkurencieschopnosti podnikov. LOGIS je odvetvovo špecializovaný predovšetkým na prostredie s diskretnou výrobou (strojárstvo, automobilový priemysel) a hutníctvo.

Dodávky sú realizované formou projektov, ktorých cieľom je zvyšovať v podnikoch zákazníkov dodávateľskú výkonnosť (byť spoľahlivým, rýchlym a pružným dodávateľom) a súčasne zlepšovať prevádzkovú efektívnosť (nízke náklady, vysoká prietoknosť).

Pre dodávky LOGISu sú charakteristické:

- **Vysoko výkonné technológie** - Technológie používané spoločnosťou LOGIS umožňujú podľa nezávislých rešeršných spoločností dosiahnutie najhodnotnejších zlepšení kľúčových výkonových parametrov podniku (tzv. KPI - Key Performance Indicators). Napr. podľa štúdií AMR sa najčastejšie podieľajú na dosiahnutí najhodnotnejších zlepšení procesov u zákazníkov celosvetovo.
- **Vysoko výkonný expertný tím** - LOGIS ako prvá spoločnosť v strednej Európe začala už pred desiatimi rokmi systematicky budovať expertný tím zameraný na uplatnenie pokročilých plánovacích technológií. Členovia tímu realizovali množstvo úspešných projektov v ČR aj v zahraničí. Pre svoje kvality sú pravidelne zapojení do projektov v zahraničí.

Vďaka tomu je LOGIS vo svojom odbore doposiaľ jedinou českou firmou, pre ktorú je dosiahnutie hmatateľných kľúčových výkonových parametrov (KPI) bežným výsledkom projektu, nie náhodnou, či výnimočnou udalosťou.

Všetky aplikácie vyvíjané firmou používajú jednotné technické prostriedky:

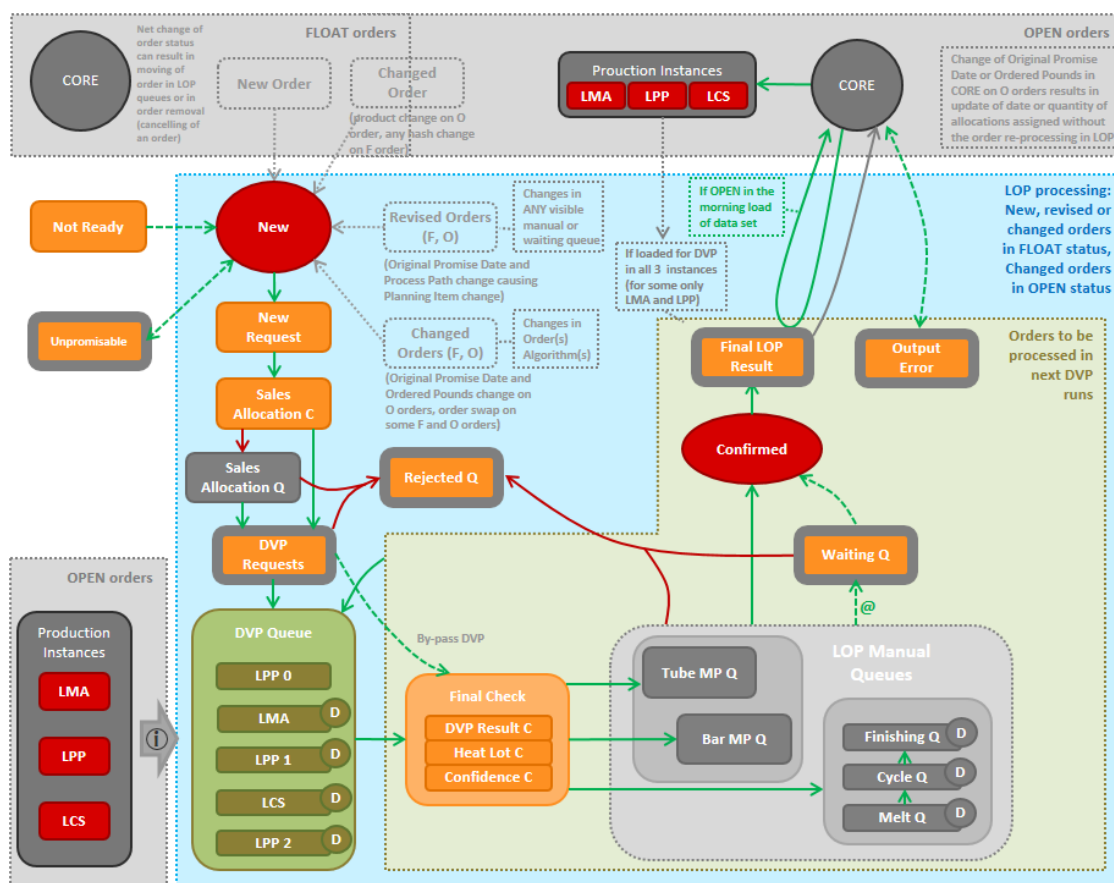
- Serverové časti sú vyvíjané v jazyku C++, klientské v Jave.
- GUI je postavené na knižnici SWING.
- Firma má implementované vlastné jadro, ktoré obsahuje všetky funkcie a kostru (ako základ) pre všetky vyvíjané aplikácie.
- Rôzne algoritmické problémy, zmeny v GUI sa vykonávajú pomocou výrazov z **LOGIS Expression Library** (kap. 2.2).

2.2 Charakteristika programu LOGIS Order Promiser

Tento program pomáha odberateľskej firme zistiť (na základe vstupných požiadavok), či sú objednávky v ich systéme splniteľné v požadovanom dátume (s požadovaným množstvom a parametrami výroby), alebo nie. Program spolupracuje so systémom v odberateľskej firme (tzv. CORE), z ktorého vždy raz za deň získava objednávky a ďalej s nimi manipuluje.

Po štarte serveru a následnom štarte klienta sa vykonáva akcia `auto_queue_processing`, ktorá objednávky spracováva a posúva medzi frontami. Každá objednávka (pokiaľ je v systéme nová) prechádza určitými frontami.

V programe existujú manuálne a automatické fronty. Ak objednávka skončí v manuálnej fronte, musí s ňou ďalej manipulovať zamestnanec, ktorý ju buď potvrdí, alebo vykoná v objednávke zmeny, prípadne ju zamietne. Detailný vývojový diagram je znázornený na obrázku č. 1.



Obr. 1: LOP Workflow

LOP sa skladá z dvoch častí:

- Server - pri spustení serveru dochádza k načítaniu dát z SQL databázy. V operačnej pamäti počítača sa vytvorí model dát, s ktorým naďalej pracuje klient (ak si vyžiada dáta). Serverové aplikácie sú teda náročné na operačnú pamäť - preto sa na testovacích / produkčných počítačoch dodávajú pamäte RAM o veľkosti minimálne 8 GB.
- Klient - komunikuje so serverom, ktorý vykonáva všetky potrebné akcie (funkcie), takže klient slúži iba ako rozhranie nad serverom (jedná sa o klasický prípad tenkého klienta).

Medzi vstupné požiadavky, pomocou ktorých sa pri spracovávaní objednávky určuje jej splniteľnosť / nesplniteľnosť, patria najmä:

- Alokácie - firma ma na každý týždeň v roku určité množstvo produktu (v tonách, resp. librách), ktoré dokáže vyprodukovať. Program má v sebe zabudovanú funkcionality nazývanú CUBE, ktorá slúži na uchovávanie týchto množstiev a pomocou ktorej sa zisťuje pri spracovávaní objednávky množstvo alokácie k dátumu, ku ktorému sa objednávka viaže.
- Dátumy - vždy sa pracuje s určitým časovým horizontom, takže ak sa spracováva objednávka, ktorá sa má splniť mimo definovaný časový horizont, stane sa "Unpromisable", teda nesplniteľná.
- DCN - Každý zákazník má pridelené svoje alokácie a akonáhle pri spracovávaní novej objednávky alokácie nie sú dostupné, zostane objednávka v manuálnej fronte Sales Allocation - táto fronta je záchytné miesto pre objednávky, ktoré nevyhoveli alokačnej požiadavke a čaká sa na manuálnu akciu (užívateľ programu ich môže potvrdiť / zmeniť / zamietnuť).

2.2.1 CUBE

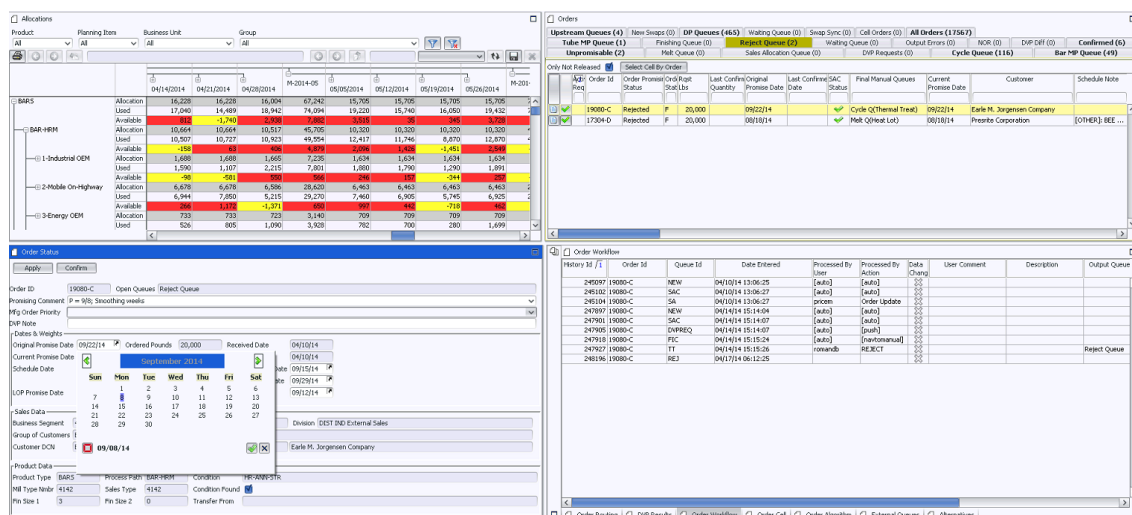
Toto označenie nesie jedna z najdôležitejších častí **LOP**. Jedná sa o spôsob, akým sa zaznamenávajú a zobrazujú alokácie na každý týždeň v roku. V jednoduchosti by sa to dalo prirovnať tabuľke v programe Microsoft Excel - s tým rozdielom, že CUBE v sebe obsahuje niekoľko druhov dát a tým pádom je viac rozmerná. Odtiaľ pochádza označenie CUBE - podľa jej rozmerovosti. Administrátor má právo meniť alokácie a tým pádom ovplyvňovať celý tok objednávok.

Každý segment (aplikácia má 3 - Industrial, Mobile On-Highway a Oil & Gas), má taktiež ešte BU rezervy. To znamená, že ak by firma nemohla poskytnúť dostatočné množstvo alokácií pre určitého zákazníka na určitý dátum, môže ešte čerpať z BU rezerv (ak sú dostupné). Tieto segmenty upravujú užívatelia, ktorí majú rolu "Chief Planner" - sú to zamestnanci odberateľskej firmy, ktorí plánujú výrobu.

V aplikácii, v okne, ktoré sa nazýva "Promising Desk" táto funkcionality zaberá 25% priestoru. Ostatných 75% tvoria fronty a detaily aktuálne vybranej objednávky.

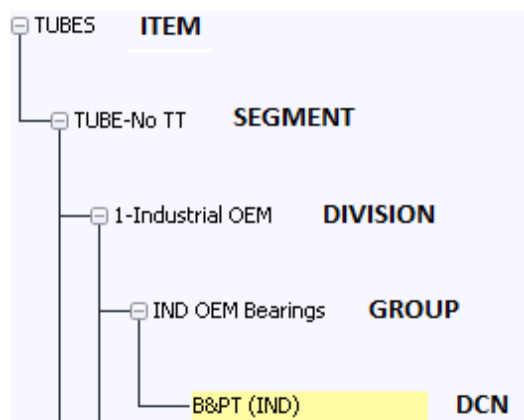
CUBE má dve osi:

- Vertikálna os - zobrazujú sa na nej jednotlivé úrovne hierarchie.
- Horizontálna os - zobrazujú sa na nej jednotlivé časové obdobia (roky, mesiace, týždne).

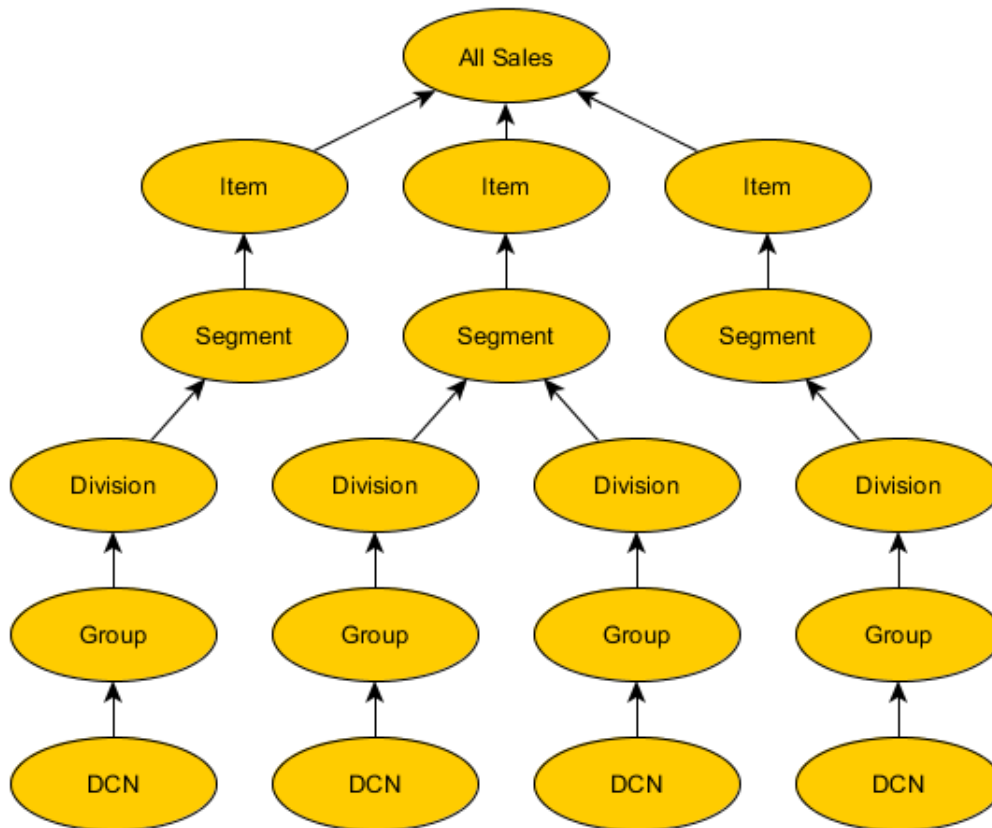


Obr. 2: Pôvodné rozloženie GUI prvkov v LOP

Hierarchia jednotlivých úrovní v tejto kocke sa dá zakresliť pomocou jednoduchého stromu.



Obr. 3: Náhľad do CUBE v reálnom použití



Obr. 4: Znázornenie CUBE ako stromu

2.2.2 Dôležité serverové akcie

- Save ICL Data
 - Vykonáva sa 1x za deň (vždy v noci) – táto serverová akcia ukladá všetky zmeny vykonané počas predchádzajúceho dňa v serverovom modeli dát do SQL databázy. Trvá približne 15 minút.
- Auto Queue Processing
 - Vykonáva sa v určitých intervaloch (dá sa nastaviť v programe, pôvodne nastavená na 1 minútu) na pozadí počas behu – táto serverová akcia spracováva objednávky a presúva ich medzi frontami.
- Auto Save
 - Vykonáva sa v určitých intervaloch na pozadí počas behu – táto serverová akcia ukladá do databázy priebežne rozdielové dáta, aby v prípade spadnutia aplikácie / serveru nedošlo ku stratám.

2.2.3 LOGIS Expression Library

Je to knižnica výrazov (s preddefinovanými premennými), pomocou ktorých je možné meniť beh programu - napríklad algoritmy, ktoré spracovávajú objednávky vo frontách. Dá sa taktiež meniť istá časť GUI nastavení (zvýrazňovanie políček). Táto knižnica obsahuje výrazy, ktoré majú byť pre odberateľskú firmu verejne dostupné - nie sú tu teda zahrnuté časti serveru, ktoré sú citlivé na zmeny.

2.3 Detail Verification Process (DVP)

Vo firme LOGIS a.s. sa momentálne vyvíja proces, ktorý spája všetky produkty vyvíjané vo firme do jedného väčšieho celku, kde všetky programy spolupracujú medzi sebou.

Ak si zoberieme klasický prípad spracovania objednávky v LOP, tak začína vo fronte **New**, odkiaľ ďalej podľa definovaných algoritmov putuje do príslušných frónt. Pri rozhodovaní, či je možné objednávku splniť v požadovanom termíne, alebo nie, využíva program niekoľko kritérií. Tieto kritériá sa ale týkajú iba softwaru LOP.

DVP tento nedostatok odstraňuje tým, že spája niekoľko softwarov do jedného celku, v ktorom objednávky putujú medzi aplikáciami tak, aby sa na konci dosiahlo presného výsledku splniteľnosti objednávky.



Obr. 5: Postupnosť, akou prechádzajú objednávky počas DVP procesu

DVP beh sa zapína manuálne v programe LOP, kde je po úspešnom behu taktiež možné výsledky zobrazíť. Avšak v určitých prípadoch je to nepraktické, preto sa vyvíja aplikácia, nazývaná **LOGIS Matrix (LMX)**, ktorá ponúka všetky funkcie (+ prehľady DVP behov) týkajúce sa tohto procesu.

3 Priebeh praxi

Táto kapitola popisuje celý môj pobyt vo firme behom praxi a všetky úlohy, ktoré mi boli pridelené. Nachádza sa tu tiež kompletný harmonogram a zoznámenie sa s firmou.

3.1 Zoznámenie sa s firmou LOGIS a.s.

Po príchode mi bolo pridelené miesto v kancelárii s počítačom. Následovalo vytvorenie účtov do e-mailového klienta a do **Tracu**. Potom som začal so samostatným prezeraním a spoznávaním prostredia softwaru *LOP* (kap. 2.2), ku ktorému mi boli poskytnuté elektronické dokumentácie *LOP User Guide* [1] a *LOP Adminsitrator Guide* [2]. Dokumentácie som si musel preštudovať a následne skontrolovať, či sa užívateľské rozhranie popísané v dokumentácii zhoduje s aktuálnou verziou softwaru.

Vďaka tejto úlohe začalo moje zoznamovanie s prostredím **LOP** a praktické skúšanie funkčnosti. Postupným prechádzaním dokumentácií spoločne s programom boli zistené určité nedostatky. Týkali sa nastavení veľkostí niektorých okien v programe. Okná boli príliš úzke, takže užívateľ nemohol vidieť potrebný obsah. Pre tieto okná boli navrhnuté nové vhodnejšie veľkosti a následne predané kolegyni.

Zistilo sa, že je potreba aktualizovať niektoré obrázky a graf popisujúci základnú logiku programu. Nedostatky boli taktiež predané kompetentnej osobe.

3.2 Harmonogram praxi

- 26.9.2013 - 27.9.2013
Prvé dni prebiehalo zaučovanie a zoznamovanie sa s prostredím programu *LOP*, kde sa testovala funkčnosť GUI prvkov.
- 3.10.2013 - 4.10.2013
Začalo sa s testovaním logiky programu a objavovaním prvých nedostatkov.
- 10.10.2013 - 11.10.2013
Bol som zoznámený so zložitejšími funkciami programu, zúčastnil som sa web meetingu.
- 17.10.2013 - 18.10.2013, 24.10.2013 - 25.10.2013
Prebiehala tvorba dokumentu, ktorý sa bude používať pri testovaní *LOP*.
- 31.10.2013 - 1.11.2013
Odovzdával som *LOP Basic Testing Scenario v0.1*, bola objavená závažnejšia chyba vo vyberači dátumov.
- 7.11.2013 - 8.11.2013, 11.11.2013
Bol vydaný nový release candidate softwaru *LOP*, takže prebiehalo dôkladné testovanie funkčnosti *LOP*.
- 14.11.2013 - 15.11.2013
Pomáhal som s testovaním užívateľských práv v software *LMP*.

- 21.11.2013 - 29.11.2013
Boli objavené chyby pri simulácii zmien na objednávke a pri kontrole dostupnosti alokácií.
- 5.12.2013 - 6.12.2013
Bolo nahlásených niekoľko chybných položiek v menu a objavený formulár, ktorý sa dostal do LOP nedopatrením (nepatril do LOP).
- 13.2.2014 - 14.2.2014
Prebiehalo testovanie nového kandidáta softwaru LOP (LOP v11 - RC1).
- 17.2.2014 - 18.2.2014
Zúčastnil som sa porady ohľadne testovania a dokončenia novej verzie LOP. Druhý deň sa konal web meeting, ktorého som sa taktiež zúčastnil.
- 20.2.2014 - 28.2.2014
Na žiadosť hlavného vývojára softwaru LOP som testoval import štruktúry CUBE z CSV súboru.
- 6.3.2014 - 14.3.2014
Prebiehalo testovanie funkcionality "Order ID Reuse" a novej funkcie, ktorá umožňovala import alokácií pre zvolený produkt a obdobie z CSV súboru, neskôr bolo objavených niekoľko chýb v logike programu.
- 17.3.2014 - 18.3.2014
Objavil som serverovú chybu a na základe podnetov od kolegov začal s úpravami LOP Basic Testing scenario.
- 20.3.2014 - 21.3.2014
Stretol som sa s hlavným vývojárom LOP, s ktorým sme prešli všetky funkcie a určili si, čo by sme mohli do dokumentu zahrnúť.
- 27.3.2014 - 28.3.2014
Od kolegu som dostal zadanie na vývoj vyhodnocovacieho software pre DVP behy.
- 3.4.2014 - 17.4.2014
Prebiehalo konzultovanie postupu práce a výslednej požadovanej štruktúry programu. Následne začal vývoj softwaru. Naprogramovanú časť aplikácie som odo vzdal a odlaďoval chyby.

3.3 Zaučovanie do programu

Po bližšom zoznámení s GUI prvkami v programe **LOP** mi bola spolupracovníkom podrobne vysvetlená celková funkcionálna a logika daného programu. Na moje poznámky a otázky mi vždy po vysvetlení aktuálnej funkcie bolo zodpovedané.

3.4 Testovanie LOP

Táto podkapitola je rozdelená do ďalších častí, ktoré opisujú úlohy vykonávané v pozícii testera. Začiatok je venovaný opisu testovania GUI prvkov, ktoré následne prechádza do logiky aplikácie. V posledných častiach boli vybrané niektoré vážnejšie chyby a následne aj detailne rozpísané.

3.4.1 GUI

V priebehu mojej praxi som bol niekoľko krát poverený otestovaním funkčnosti GUI prvkov. Väčšinou sa jednalo o testovanie nových funkcií, prípadne o testovanie opravených chýb. Do kontroly patrili napríklad nasledujúce funkcie:

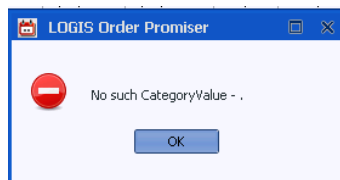
- Tabuľkové filtre
- Zobrazenia jednotlivých polí (v tabuľkách), ktoré sú uložené v SQL databázi
- Ukladanie užívateľských nastavení programu (a opätovné načítanie po reštarte)
- Vlastné *Java Swing* komponenty

3.4.2 Logika programu

Ďalšou úlohou bolo testovanie logiky programu, ktoré bolo rozdelené na nasledujúce časti:

- Kontrolovanie toku objednávky (tzv. workflow) - zistiť, či každá objednávka prechádza správnymi frontami kontrolujúcimi možnosti jej realizácie (kap. 2.2).
- Tlačítka spúšťajúce určité akcie (schválenie objednávky, odmietnutie a pod.) - zistiť, či sa pri spustení danej akcie vykonajú patričné zmeny v objednávke, napríklad ak spustíme akciu "odmietnutie objednávky", tak by sa mala objednávka presunúť do fronty "Rejected Queue".
- Kontrola časti programu nazývanej CUBE - táto časť je opísaná v kapitole 2.2. Bolo potrebné zistiť, či sa dajú ukladať zmeny v alokáciách a či sa správne zobrazujú jednotlivé pohľady.

Pri testovaní týchto jednotlivých častí bolo nájdených niekoľko chýb, ktoré som sa snažil riešiť. Niektoré chyby sa týkali nepresných textov v chybových hláškach (štandardné Message boxy v programovacom jazyku **Java**). Jedna chyba sa týkala zberu dát z SQL databázy a nesprávne interpretovanie v **LOP**.



Obr. 6: Chybová hláška - No such category value.

Vyššie uvedená chybová hláška vyskakovala vtedy, keď chcel užívateľ v programe simulovať dopad zmien v objednávke (napr. zmena dátumu, kedy by mala byť objednávka spracovaná, zmena rozmerov výrobku, prípadne zmena požadovanej hmotnosti výrobku), ktorá mala stav "Unpromisable" - tento stav označoval nesplniteľné objednávky.

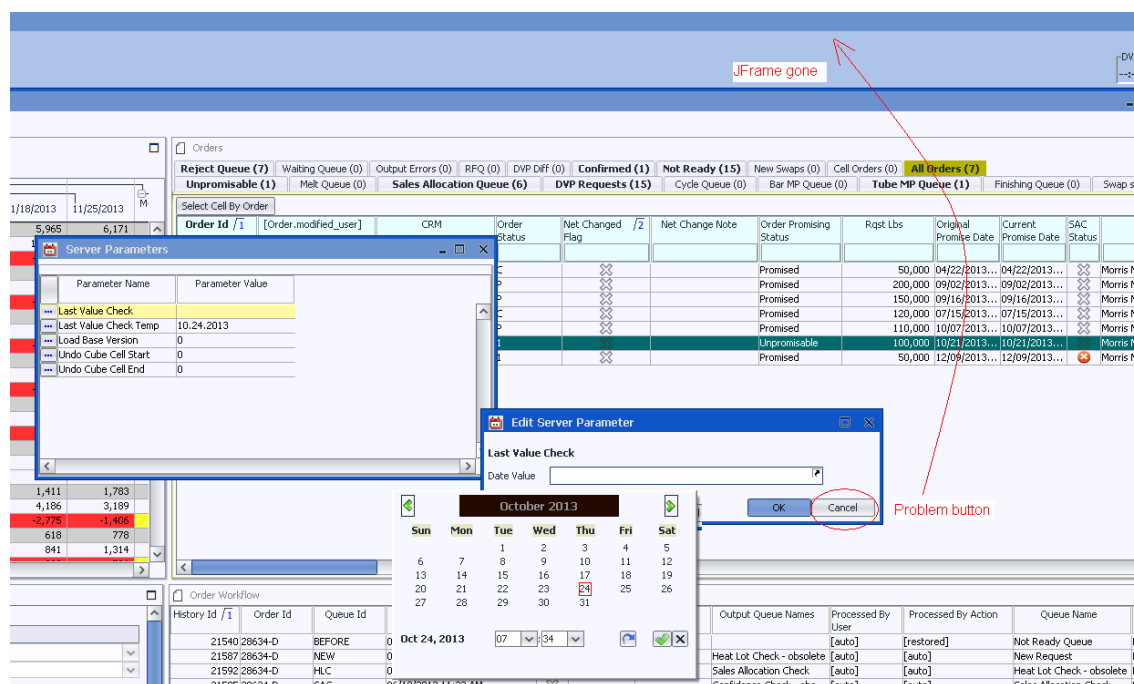
Zistilo sa, že bol problém v získavaní dát z SQL servera (konkrétne v dopyte, ktorý vyberal z tabuľky informácie o objednávke). Do SQL dopytu sa dostalo ID nesplniteľnej objednávky a keďže tieto objednávky nemajú žiadne iné záznamy v tabuľkách obsahujúcich napríklad alokácie alebo alternatívy, tak pri dohľadávaní potrebných údajov pre simulovanie zmien boli z dopytu vrátené prázdne výsledky, čo viedlo k vyššie uvedenej chybe.

Riešenie spočívalo v pridaní kontroly, či sa jedná o splniteľnú alebo nesplniteľnú objednávku.

3.4.3 Chyba so zmiznutím hlavného JFrame pri výbere dátumu

Ak chcel užívateľ zmeniť dátum v nastaveniach pomocou vyberača dátumu a klikol na tlačítko "Cancel", ktoré sa nachádzalo vo formulári na pozadí (rodič vyberača), zmizol hlavný JFrame (konkrétne nadpis, ikonka, tlačítka na maximalizovanie, minimalizovanie a zavretie okna) - detailné znázornenie nájdete na obrázku č.7.

Moje zistenia boli predané kompetentným osobám z vývojového tímu (napísaním nového hlásenia chyby do Tracu).



Obr. 7: Zmiznutie hlavného JFrame

3.4.4 Chyba pri používaní funkcionality "One Order Algorithm"

Táto funkcionality slúži na simulovanie zmien v objednávke (užívateľ môže niektoré objednávky zmeniť parametre ako množstvo, dátum a pod.) a ako výsledok tejto operácie by mal byť dátum, kedy by takto zmenená objednávka bola splniteľná. Ak užívateľ ale zo zoznamu vybral nespľniteľnú objednávku, objavila sa chybová hláška, ktorá vypisovala "No such Category Value".

Chyba bola nahlásená a odovzdaná kompetentnej osobe na opravenie. Po opravení bola chyba znova otestovaná a po zistení, že bola opravená bola označená za splnenú.

3.4.5 Ukladanie alokácií po vykonaní akcie "Save ICL Data"

Zistilo sa, že sa neukladajú zmeny v alokáciách po vykonaní akcie "Save ICL Data" (kap. 2.2). Tento problém sa objavil v systéme ako opravený a tak bolo potrebné ho dôkladne otestovať. Táto akcia neukladala zmeny vykonané užívateľom v časti nazvanej CUBE, kde užívateľ menil alokácie (tj. koľko ton produktov je schopná firma vyprodukovať za určitý týždeň / mesiac).

Testovanie sa vykonávalo tak, že sa manuálne zmenili alokácie a spustila sa akcia "Save ICL Data". Po dokončení akcie sa reštartoval server. Predtým, ako sa pri reštarte spustil znovu server, sa musela ešte vykonať uložená SQL procedúra "LOP_Daily_Cleanup" - táto procedúra sa spúšťa automaticky na serveri každé ráno, aby sa vyčistili informácie o verzii z predchádzajúceho dňa. K týmto záverom (a úspešnému vykonaniu tejto procedúry)

som dospel sám a využil som znalosti z predmetov Úvod do databázových serverov a Databázové a informačné systémy (konkrétne ovládanie programu SQL Management Studio, porozumenie SQL procedúram). Po načítaní serveru boli porovnané alokácie a zapísané výsledky. Zistilo sa, že problém bol úspešne opravený a tak sa označil tento problém za vyriešený.

3.4.6 Testovanie funkcionality "Change Open Order Attributes"

Funkcionalita slúži na vykonávanie zmien (množstvo, dátumy, rozmery výrobkov) v objednávkach, ktoré sú otvorené (nachádzajú sa v LOP spracovávaní). Test prebiehal tak, že bola vybraná objednávka a vykonali sa na nej zmeny.

V objednávke sa vykonali nasledovné zmeny:

- Original Promise Date & Quantity - Pri zmene dátumu alebo množstva by sa mal stav objednávky zmeniť na „New“ a objednávka by mala znova vstúpiť do spracovávania (tj. prechádzanie všetkými frontami).

Po vykonaní zmien sa zistilo, že vyššie popisované funkcie sa vykonávajú korektne.

3.5 Web meeting

Firma uskutočňuje každý týždeň s americkou firmou **Timken** web meeting, kde sa preberajú firemné veci:

- v akom stave sú vyvíjané softwarové riešenia.
- plánovanie nových funkcionalít.
- plánovanie služobných ciest do Ameriky.
- vyskytnuté problémy pri používaní softwaru v prevádzke.

Jedného web meetingu som sa zúčastnil a bol pre mňa veľmi užitočný, pretože som mohol sledovať, ako prebieha komunikácia medzi svetovými firmami v anglickom jazyku, taktiež som vďaka nemu pochopil, kto akú rolu pri tvorbe firemných softwarov hrá a že každý článok tímu je vo svojej podstate rovnako dôležitý ako správna komunikácia medzi nimi.

3.6 Tvorba testovacieho scenára

Jednalo sa o vyhotovenie dokumentu, ktorý bude slúžiť testerom (vo firme, alebo u odberateľa) softwaru **LOP**. Dokument bude obsahovať základné body, ktoré bude potreba otestovať vždy, keď vývojový tím vydá novú verziu **LOP**. Dokument bol písaný v anglickom jazyku.

Dokument bol rozdelený na dve časti:

- Simplified version - zjednodušená verzia, bude využívaná skúsenými testerami vo firme.
- Detailed version - detailná verzia, bude využívaná neskúsenými (novými) testerami vo firme, prípadne u klienta. Obsahuje detailne popísané scenáre, ktoré by sa mali otestovať.

Na vyhotovenie mi boli stanovené dva týždne, resp. 4 pracovné dni. Po prečítaní podkladov k úlohe bolo načrtnuté základné rozdelenie dokumentu. Postupne sa začal prechádzať program a zisťovať, ktoré veci by sa mohli do testovania zahrnúť.

Prechádzaním programu sa narazilo na niektoré funkcie, ktoré boli pre mňa cudzie, takže som potreboval rady odborníka. Svoje otázky som smeroval teda na kolegu z testovacieho tímu. Po objasnení som zhodnotil, či by mala daná funkcionálna patriť do testovacieho scenára, alebo nie.

Po dokončení sa základné rozdelenie dokumentu pozmenilo a nakoniec sa dospelo k nasledujúcej štruktúre:

- GUI - prvky užívateľského rozhrania.
- Promising Desk - kontrola funkcií v hlavnom okne programu.
- Orders Swap - funkcie týkajúce sa výmeny objednávok.
- Change Open Order Attributes - zmena atribútov otvorenej objednávky.

Po niekoľkých týždňoch bolo treba dokument prerobiť a to z toho dôvodu, že sa mal poslať do odberateľskej firmy, kde by ho mali interní zamestnanci používať pri ich testovacom procese.

Z pôvodnej štruktúry sa dokument rozšíril nasledovne:

- Allocations
- Queues
- Assignments
- Reports
- Swaps
- Change Open Orders Attributes

- Quick Sales Allocation Check
- User Rights
- E-Mails
- GUI
- NOR

3.7 Testovanie užívateľských práv v LMA

Bolo potrebné otestovať užívateľské práva v softwari **LMA**. Aplikácia môže bežať v dvoch prostrediach:

- PROD - používa sa v prevádzke.
- DVP - používa sa pri behu DVP (kap. 2.3).

Tieto prostredia sa dajú definovať v konfiguračnom súbore LMP.cfg a to nasledujúcim spôsobom:

```
set-property instance_type:PROD
set-property instance_type:DVP
```

Výpis 1: Nastavovanie prostredí v LMA

Postupne sa museli skúšať oboje prostredia. Testovali sa základné užívateľské práva pre určité prostredie. To znamená, že ak pri pridávaní práva užívateľovi nastavíme jeden z nižšie uvedených výrazov, bude mať nastavené právo dostupné iba pre vybrané prostredie.

```
in(instance_type,"PROD")
in(instance_type,"DVP")
```

Výpis 2: Výraz pre obmedzenie práv na určité prostredie

Výsledky ukázali, že tieto práva fungujú na oboch prostrediach korektne.

3.8 Tvorba dokumentácie k softwaru LMP

Bol som požiadaný o vyhotovenie dokumentácie k ďalšiemu produktu od firmy **LOGIS a.s.**, a to LOGIS Master Planner. Je to nový produkt (ktorý je stále vo vývoji), slúžiaci na plánovanie smien vo firme. Jednoduché vysvetlenie (ako program funguje), mi bolo poskytnuté kolegom, ktorý software vyvíja. V podstate sa jednalo o vytvorenie kostry dokumentácie, pretože software ešte neumožňoval skoro žiadne špecifické funkcie. Začalo sa opisovaním základného ovládania programu, potom som pokračoval popisovaním filtrov a ku všetkému som doplnil obrázky. Dokumentácia musela byť celá v anglickom jazyku.

3.9 Import štruktúry CUBE z CSV súboru

Bol som požiadaný o testovanie novej funkcie, ktorú práve implementoval hlavný člen vývojového tímu. Jednalo sa o načítanie štruktúry CUBE z CSV súboru.

Cieľom tohto testovania boli nasledujúce body:

- **Načítanie CSV súboru**

- Sem patrilo otestovanie chybových hlášok v prípade, že je na vstupe CSV súbor v nesprávnom tvare. Muselo sa zaistiť, že LOP nespadne pri vyberaní chybného súboru.

Patrili sem tieto chybové hlášky:

- * **Kontrola správneho CSV súboru** - ak by bol na vstupe súbor, ktorý má iné názvy jednotlivých stĺpcov ako potrebujeme, hláška by mala informovať o tejto udalosti.
- * **Prázdny CSV súbor**
- * **Chýbajúce stĺpce v CSV súbore** - súbor musí obsahovať všetky z nižšie uvedených stĺpcov
 - CATEGORY_VALUE_ID, CATEGORY_LEVEL_ID, PARENT_CATEGORY_VALUE_ID, CATEGORY_VALUE_NAME
- * **Názvy stĺpcov nie sú v prvom riadku**
- * **V stĺpci CATEGORY_LEVEL_ID budú chybné hodnoty** - stĺpec musí obsahovať iba niektoré z nižšie uvedených hodnôt
 - ALL-PRODS, ALL-SALES, ALL-TIME, DCN, DIV, GRP, ITEM, MON, PROD, QTR, SEG, WEEK, YEAR
- * **Prehodené poradie stĺpcov / riadkov** - program musel korektne spracovať aj súbor, ktorý mal prehodené riadky, alebo stĺpce.

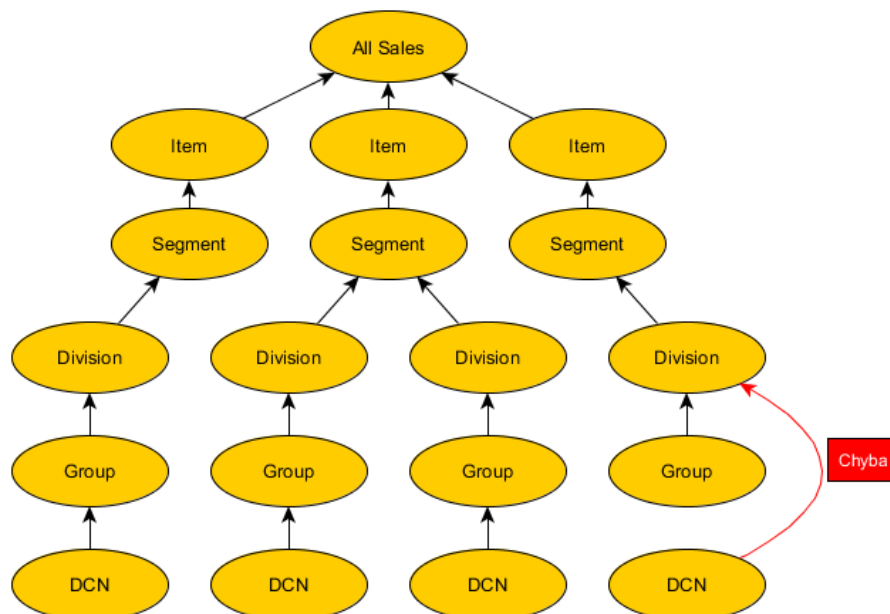
- **Správne uloženie novej štruktúry do databázy**

- Musí odpovedať štruktúra (kap. 2.2).
- Každý potomok musí mať svojho rodiča (pokiaľ sa nejedná o koreň, ktorý nemá rodiča).
- Bezchybné potvrdenie uloženia štruktúry.
- Po reštarte serveru korektné načítanie novej štruktúry - porovnanie vybraných dát z CSV súboru so zobrazenou štruktúrou CUBE v LOP.

Po riadnom otestovaní vyššie uvedených bodov som zistil niekoľko nedostatkov:

- Chýbala hláška pri vybraní prázdneho CSV súboru.
- Algoritmus nesprávne vyhodnotil správnosť štruktúry - ignoroval podmienku, že každý potomok musí mať svojho rodiča, takže prepúšťal aj potomkov bez rodičov.
- Algoritmus prepúšťal chybnú štruktúru - v tejto štruktúre sa viazal potomok na rodiča, ale cez 2 úrovne vyššie (obr. 8).

Všetky chyby boli predané kompetentnej osobe na ďalšiu analýzu a následnú opravu.



Obr. 8: Zlá štruktúra, ktorú LOP vyhodnotil ako správnu

3.10 Tvorba softwaru pre vyhodnocovanie výsledkov DVP behu

Pre efektívnejšie vyhodnocovanie výsledkov DVP behu sa začala vo firme vyvíjať aplikácia, ktorá si po dokončení DVP behu získa z SQL databázy záznamy zo všetkých nástrojov a zobrazí ich vo formáte HTML stránky.

Pri vývoji tohto softwaru som pomáhal s časťou, ktorá sa týkala **LOP**. V LOP časti mal software vyhodnocovať nasledujúce prípady:

- Počet objednávok, ktoré vstúpili (prišli z LOP) do DVP procesu.
- Počet objednávok, ktoré sa po skončení DVP procesu neobjavili v manuálnych frontách (boli nesprávne spracované).
- Počet objednávok, ktoré sa vrátili z DVP procesu.
- Počet objednávok, ktoré nemajú pridelené žiadne alokácie (predpokladaný maximálny počet je 10% z počtu objednávok, ktoré vstúpili do DVP procesu).

Počet objednávok, ktoré vstúpili (prišli z LOP) do DVP procesu

Objednávky, ktoré vstúpili do DVP procesu z LOP sa nachádzajú v tabuľke LOP_DEMANDORDER.TO.DVP. Táto tabuľka obsahuje informácie o objednávke. Ako príklad uvediem najpoužívanejšie:

- ORDER_ID
- PROMISE_DATE_ORIG
- LOP_ORDER_STATUS
- QUANTITY

V tomto kroku sa využil stĺpec LOP_ORDER_STATUS, aby sa do SQL dopytu nezahrňovali objednávky, ktoré majú stav "Rejected" - teda zamietnuté.

```
SELECT COUNT(*)
FROM LOP_DEMANDORDER.TO.DVP
WHERE LOP_ORDER_STATUS != 'Rejected'
```

Výpis 3: Výsledný SQL dopyt pre spočítanie objednávok, ktoré vstupujú do DVP procesu

Po navrhnutí SQL dopytu bolo zahájené testovanie v programe Microsoft SQL Management Studio. Po úspešnom otestovaní sa tento dopyt použil v aplikácii. Výsledok SQL dopytu bol uložený do premennej typu integer.

Počet objednávok, ktoré sa vrátili z DVP procesu

Pri navrhovaní SQL dotazu pre získanie počtu objednávok, ktoré sa vrátili z DVP procesu, bolo potrebné rozdeliť úlohu na dve časti:

- Získanie ID posledného DVP behu
- Získanie počtu objednávok z posledného DVP behu

Získanie ID posledného DVP behu:

Všetky záznamy DVP behov sú uložené v tabuľke DVP_RUN. Z tabuľky bolo potrebné získať primárny kľúč (DVP_RUN_ID) u najnovšieho záznamu. Keďže PK sa pričítava inkrementačne, použil som SQL funkciu MAX().

```
SELECT MAX(DVP_RUN.ID)
FROM LOP_DVP_RUN
```

Výpis 4: SQL dopyt pre získanie najnovšieho DVP behu

Výsledok dopytu som si uložil do premennej pre ďalšie použitie.

```
SELECT COUNT(*)
FROM LOP_DEMANDORDER_FROM_DVP
WHERE DVP_RUN_ID=lastDVPRunId
```

Výpis 5: Výsledný SQL dopyt pre získanie počtu objednávok, ktoré sa vrátili z DVP procesu

Počet objednávok, ktoré nemajú pridelené žiadne alokácie

Keďže aj v tomto prípade je potrebné ID DVP behu, bude použité z predchádzajúceho SQL dopytu. Aby mohla byť považovaná objednávka za takú, ktorá nemá priradené žiadne alokácie, je potrebné, aby mala nasledujúce stĺpce v tabuľke LOP_DEMANDORDER_FROM_DVP prázdne:

- available_material_alloc
- heat_allocation
- new_heat

```
SELECT COUNT(*)
FROM LOP_DEMANDORDER_FROM_DVP
WHERE isnull(available_material_alloc,0)=0
AND isnull(heat_allocation,0)=0
AND isnull(new_heat,0)=0
AND DVP_RUN_ID=lastDVPRunId
```

Výpis 6: Výsledný SQL dopyt pre získanie počtu objednávok, ktoré nemajú žiadne alokácie

Všetky výsledky z SQL dotazov boli následne zapísané do HTML súboru. Ukážkový prehľad z výsledkov jedného DVP behu je zobrazený na obrázku 9.

Actual Value			Expected Value
73	LMX	Workflow steps run	73
74	LPP	Number of History Records	min 32
319	LOP	DVP Orders	
316	LOP	Post DVP queues - !!! Is current LOP data saved?	319
319	LOP	DVP Results	319
12	LOP	No Allocation	max 31

Obr. 9: Výsledný prehľad DVP výsledkov

4 Záver

Moja bakalárska prax splnila moje očakávania. Chcel som si vyskúšať, aké je to pracovať vo firme, v ktorej je dôležitá koordinácia spolupracovníkov a časová dôslednosť. Tím vo firme je veľmi dobre organizovaný, takže hravo zvládajú aj náročné úlohy. Dozvedel som sa, ako prebieha vývoj a testovanie, aké technické prostriedky sa používajú na zaznamenávanie chýb (systém Trac) a akými fázami prebieha vývoj až do samotného odoslania do odberateľskej firmy.

Boli mi pridelené úlohy, ktoré som vyriešil v požadovanom časovom rozmedzí. Počas praxe som sa podieľal na vývoji vyhodnocovacieho softwaru pre výsledky z DVP behov, prispel som svojim testovaním k vyladeniu vyvíjaných aplikácií (prevažne LOP), v anglickom jazyku som tvoril testovacie scenáre a užívateľské príručky pre potreby firmy. Pri testovaní som navrhoval zmeny v užívateľskom rozhraní LOP, pri nahlasovaní problémov som k opisu problému vždy pridával mnou navrhnuté riešenie problému. Musel som si naštudovať dokumentácie LOP User Guide [1], LOP Administrator Guide [2] aby som sa zoznámil s aplikáciou LOP.

Pri praxi som využil predovšetkým svoje znalosti z predmetov Úvod do databázových systémov a Databázové a informačné systémy, ktoré mi uľahčili prácu s programom Microsoft SQL Management Studio, naučili ma jazyk SQL, ktorý som využil pri písaní dopytov do vyhodnocovacieho softwaru. V neposlednom rade mi bol veľkým prínosom predmet Úvod do softwarového inžinýrství, ktorý som využil pri tvorbe vývojových diagramov, ďalej tiež Programovacie jazyky I a Užívateľské rozhrania. Pri komunikácii v anglickom jazyku mi pomohli odborné termíny z predmetu anglického jazyka. Pri návrhu zmien v užívateľskom rozhraní som využil znalosti z predmetu Užívateľské rozhrania, ktorý mi ukázal, ako by mala aplikácia vyzeráť, aby bola pre užívateľa príjemná na pohľad a dobre sa s ňou pracovalo.

Patrik Herák

5 Literatúra

- [1] *LOGIS Order Promiser User Guide*, LOGIS a.s., 2013
- [2] *LOGIS Order Promiser Administrator Guide*, LOGIS a.s., 2013
- [3] *Webové stránky firmy LOGIS a.s. - Firemní profil [online]*. [cit. 15.2.2014]. Dostupné z:
<http://www.logis.cz/web/page.php?26>
- [4] *W3Schools Online Web Tutorials - SQL tutorial [online]*. Dostupné z:
<http://www.w3schools.com/sql/>
- [5] *The Java Tutorials [online]*. Dostupné z:
<http://docs.oracle.com/javase/tutorial/>